

**A SYSTEM AND METHOD FOR AUTONOMIC PERFORMANCE ENHANCEMENT
OF STORAGE MEDIA**

BACKGROUND OF THE INVENTION

5

Field of the Invention

The present invention relates generally to the improvement of operation of storage media and, more particularly, to improving the storage algorithms on storage
10 media.

Description of the Related Art

Within a computer system, there are a variety of types of storage media that range from volatile and nonvolatile
15 memory, typical types of magnetic media, to new magneto-optic media. Specifically, there are a series of nonvolatile magnetic and magneto-optic storage media that utilize a head and rotating disks or platters, such as a Hard Disk Drive (HDD) or a Floppy Disk Drive. Files are
20 placed on the rotating disk at easily accessible locations depending on the configuration of the disk system. If it is a multi-platter, multi-head assembly, then the configuration can vary from drive to drive.

The majority of disk systems used today are designed to
25 provide maximum write performance. In other words, the drives are designed to write to the disk or disks at a maximum rate. In multi-platter systems, maximum write speed usually results in the utilization of all available heads

writing across different platters. Hence, storage blocks for files are not contiguous, but are scattered. As files are deleted or updated, storage blocks become even more widely scattered in a non-contiguous fashion, or fragmented.

5 This fragmentation may start from a desire for maximum write speed, but efficiency in accessing the files is decreased. The heads are required to seek across large portions of the entire disk system to locate clusters or sectors containing file fragments. The fragmentation may
10 appear to be insignificant; however, over time with a number of files written, deleted, or rewritten, files fragments can be scattered to a point where access time to files is so long as to create drastic delays.

To alleviate the problem associated with file
15 fragmentation, a common practice is to defragment the drive periodically. Defragmentation is typically initiated by a user or administrator and can require several minutes to several hours to complete. Also, defragmentation usually occurs once the fragmentation has or is causing critical
20 delays. These critical delays typically result in a loss of productivity.

Therefore, there is a need for a method and/or apparatus for improving the read and write capabilities for a disk drive that addresses at least some of the problems
25 associated with fragmentation while utilizing conventional

methods and apparatuses for the read and write capabilities of a disk drive.

SUMMARY OF THE INVENTION

5 The present invention provides a method for monitoring file defragmentation of at least one storage medium at least coupled to a computer system. A determination is made as to if file fragmentation occurs when data is written to, deleted from, or scanned from the at least one storage
10 media. Also, locations of a plurality of file fragments are stored when the system monitor has at least determined that file fragmentation has occurred in a storage medium.

BRIEF DESCRIPTION OF THE DRAWINGS

15 For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

FIGURE 1 is a block diagram depicting a platter
20 containing fragmented files;

FIGURE 2 is a block diagram depicting a defragmented platter;

FIGURE 3 is a flow chart depicting the operation of a fragmentation tracker and defragmentation agent; and

25 FIGURE 4 is a block diagram depicting file attribute measurements for use in the defragmenter.

DETAILED DESCRIPTION

In the following discussion, numerous specific details are set forth to provide a thorough understanding of the present invention. However, those skilled in the art will appreciate that the present invention may be practiced without such specific details. In other instances, well-known elements have been illustrated in schematic or block diagram form in order not to obscure the present invention in unnecessary detail. Additionally, for the most part, details concerning network communications, electro-magnetic signaling techniques, and the like, have been omitted inasmuch as such details are not considered necessary to obtain a complete understanding of the present invention, and are considered to be within the understanding of persons of ordinary skill in the relevant art.

It is further noted that, unless indicated otherwise, all functions described herein may be performed in either hardware or software, or some combinations thereof. In a preferred embodiment, however, the functions are performed by a processor such as a computer or an electronic data processor in accordance with code such as computer program code, software, and/or integrated circuits that are coded to perform such functions, unless indicated otherwise.

Referring to FIGURE 1 of the drawings, the reference

numeral 100 generally designates a platter containing fragmented files. Within the platter 100, there typically can be several thousand to several billion or more clusters (not shown). These clusters typically store a small amount of data magnetically. Hence, a file typically cannot be stored in a single cluster, but in multiple clusters. For example, File 1 Clusters 110 and File 2 Clusters 120 are spread across the platter. The unordered pattern of the File 1 Clusters 110 and the File 2 Clusters 120 denotes file fragmentation.

Referring to FIGURE 2 of the drawings, the reference numeral 200 generally designates a defragmented platter. Also, within the platter 200, there typically can be several thousand to several billion or more clusters (not shown). However, in a defragmented platter, file clusters are ordered into a specific pattern. For example, File 1 Clusters 210 and File 2 Clusters 220 are not spread across the entire platter, but are precisely ordered.

Referring to FIGURE 3 of the drawings, the reference numeral 300 generally designates a flow chart depicting the operation of a fragmentation tracker and a defragmentation agent. In order to better improve efficiency of the disk drive, two distinct algorithmic creations, the fragmentation tracker and the defragmentation agent, are utilized. The fragmentation tracker is responsible for monitoring and

maintaining an account of the location of fragmented file clusters. The defragmentation agent is responsible for implementing defragmentation of the file fragmentations determined by the fragmentation tracker. The procedure for
5 the operation of the fragmentation tracker and the defragmentation agent are depicted with the flow chart 300.

Steps 302, 304, 306, and 308 depict the operation of the fragmentation tracker. In step 302, a scan, write, or delete occurs as a result of data storage operation. For
10 example, a user deletes a file. In step 304, once the scan, write, or delete has occurred, the fragmentation tracker must determine if file fragmentation has occurred. In step 306, if file fragmentation has not occurred, then no defragmentation is necessary. In step 308, if file
15 fragmentation has occurred, then the fragmentation tracker stores the location of the file fragments for future defragmentations.

If defragmentation is necessary, then in step 310, a determination of whether the system is idle must be made.
20 Typically, in conventional systems, the user would initiate a defragmentation algorithm to alleviate the problems associated with fragmentation. However, it is inefficient to wait until fragmentation causes substantial or critical delays. There would be a better use of computer resources
25 to operate defragmentation algorithms during inactive

periods that occur intermittently. Hence, in step 312, if the system is not idle, then the defragmentation agent remains dormant until the system is idle.

Once the system becomes idle, then in step 314, the file
5 is defragmented. The defragmentation then continues until the file is completely defragmented 316. Once the defragmentation is complete, then the tracker reference regarding the defragmented file is reset 318.

The agent, though, does not completely lock out usage
10 once defragmentation begins. During the defragmentation completion process 316, defragmentation can be stopped. If defragmentation is stopped before completion, there should be a determination as to whether the defragmentation has been halted as a result of activity 320. If there has not
15 been any activity, then an error is reported. If there has been activity, the agent becomes dormant and waits for the system to become idle again 312.

Referring to FIGURE 4 of the drawings, the reference numeral 400 generally designates a block diagram depicting
20 file attribute measurements for use in the defragmenter.

Depending on the disk drive type and usage, the characteristics for defragmentation of the drive vary based on the desired usage. In the typical desktop computer, when a drive is defragmented, the user has virtually no input on
25 the defragmentation algorithm. For example, the user is

typically unable to set defragmenter priorities for file types. However, systems administrators can have much more input into drive defragmentation. For example, a systems administrator can set priorities for file types. There can
5 be a variety of user-defined attributes that define the defragmentation. The most likely user defined attributes would include such settings as read speed 430, write speed 432, access frequency 404, and so forth. Typically, there are several other attributes that can be used in a
10 defragmentation, such as file size 412, typical duration of access time 406, file/application association 410, last time since accessed 408, read characteristics 414, update characteristics 416, time of day access normally occurs 418, application priority 428, and file/application type 402.
15 There may also be a variety of other attributes that contributed to defining a defragmentation 422.

Once all of the attributes have been entered into the algorithm, the defragmentation agent 424 can properly function. The inputted attributes allow for a variety of
20 desired performance enhancements. Taking into account all of the inputted attributes, the defragmentation agent then may re-associate file clusters to locations on the platter or platters that correspond to the specific, desired defragmentation 426.

It will further be understood from the foregoing description that various modifications and changes may be made in the preferred embodiment of the present invention without departing from its true spirit. This description is
5 intended for purposes of illustration only and should not be construed in a limiting sense. The scope of this invention should be limited only by the language of the following claims.